AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented method for arranging ~~computer~~ two-dimensional vector graphics data for processing into an output, the method comprising:

receiving a function call via an application programming interface of a media integration layer that is among a plurality of layers in a graphics processing environment, the function call including ~~a~~ markup language data in a native format, the markup language data comprising direct code calls, object model code calls, and XML-based markup, and the function call corresponding to graphics-related data;

a parser/translator interpreting the markup language data in its native format causing data in a scene graph data structure to be modified based on the function call; and

causing a change in a display in response to the modification of data in the scene graph.

2. (previously presented) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to initialize a new instance of a non-drawing visual class.

3. (original) The method of claim 2 further comprising, receiving a function call via an interface corresponding to a transform associated with the visual.

4. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to initialize a new instance of a drawing visual class.

5. (original) The method of claim 4 further comprising, receiving a function call via an interface to open the drawing visual instance for rendering, and in response, causing a drawing context to be returned, the drawing context providing a mechanism for rendering into the drawing visual.

6. (original) The method of claim 1 further comprising, receiving brush data in association with the function call, and wherein causing data in a scene graph data structure to be modified comprises invoking a brush function to modify a data structure in the scene graph data structure such that when a frame is rendered from the scene graph, an area will be filled with visible data corresponding to the brush data.

7. (original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a solid color.

8. (previously presented) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a linear gradient brush and a stop collection comprising at least one stop.

9. (original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a radial gradient brush.

10. (original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to an image.

11. (original) The method of claim 10 further comprising, receiving a function call via an interface corresponding to an image effect to apply to the image.

12. (original) The method of claim 1 further comprising, receiving pen data in association with the function call, and wherein causing data in a scene graph data structure to be modified comprises invoking a pen function that defines an outline of a shape.

13. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a geometry-related function to represent an ellipse in the scene graph data structure.

14. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a geometry-related function to represent a rectangle in the scene graph data structure.

15. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a geometry-related function to represent a path in the scene graph data structure.

16. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a geometry-related function to represent a line in the scene graph data structure.

17. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function related to hit-testing a visual in the scene graph data structure.

18. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function related to transforming coordinates of a visual in the scene graph data structure.

19. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function related to calculating a bounding box of a visual in the scene graph data structure.

20. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function via a common interface to a visual object in the scene graph data structure.

21. (original) The method of claim 1 further comprising invoking a visual manager to render a tree of at least one visual object to a rendering target.

22. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place a container object in the scene graph data structure, the container object configured to contain at least one visual object.

23. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place image data into the scene graph data structure.

24. (original) The method of claim 23 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place an image effect object into the scene graph data structure that is associated with the image data.

25. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place data corresponding to text into the scene graph data structure.

26. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to provide a drawing context in response to the function call.

27. (original) The method of claim 26 wherein the function call corresponds to a retained visual, and further comprising, calling back to have the drawing context of the retained visual returned to the scene graph data structure.

28. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place a three-dimensional visual into the scene graph data structure.

29. (original) The method of claim 28 wherein causing data in a scene graph data structure to be modified comprises mapping a two-dimensional surface onto the three dimensional visual.

30. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place animation data into the scene graph data structure.

31. (original) The method of claim 30 further comprising communicating timeline information corresponding to the animation data to a composition engine at another layer of the media integration layer.

32. (original) The method of claim 31 wherein the composition engine interpolates graphics data based on the timeline to animate an output corresponding to an object in the scene graph data structure.

33. (original) The method of claim 1 wherein receiving a function call via an interface of a media integration layer comprises receiving markup, and wherein causing data in a scene graph data structure to be modified comprises parsing the markup into a call to an interface of an object.

34. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place an object corresponding to audio and/or video data into the scene graph data structure.

35. (original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises changing a mutable value of an object in the scene graph data structure.

36. (previously presented) In a computing environment, a computer system comprising:

a scene graph data structure of a layered system for containing data that can be rendered into output that for subsequent integrated output that can be viewed; and

an object model including objects and other data that can be contained in the scene graph data structure, at least some of the objects of the object model having application program interfaces in a media integration layer that is among a plurality of layers in a graphics processing environment for invoking functions to modify contents of the scene graph data structure, wherein the functions are in a markup language in a native format.

37. (original) The system of claim 26 wherein at least one function is invoked to place a tree of visual objects into the scene graph data structure.

38. (original) The system of claim 37 further comprising a visual manager that when invoked renders the tree of visual objects to a rendering target.

39. (original) The system of claim 37 wherein the tree of visual objects is contained in a visual collection object.

40. (original) The system of claim 26 wherein at least one function is invoked to place a visual object into the scene graph data structure.

41. (original) The system of claim 40 wherein at least one function is invoked to associate a brush with the visual object.

42. (original) The system of claim 40 wherein at least one function is invoked to associate a geometry with the visual object.

43. (original) The system of claim 42 wherein the geometry comprises at least one of a set containing an ellipse geometry, a rectangle geometry, a line geometry and a path geometry.

44. (original) The system of claim 40 wherein at least one function is invoked to associate a transform with the visual object.

45. (original) The system of claim 44 wherein the transform comprises a rotate transform for changing a perceived angle of the visual object.

46. (original) The system of claim 44 wherein the transform comprises a scale transform for changing a perceived size of the visual object.

47. (original) The system of claim 44 wherein the transform comprises a translate transform for changing a perceived position of the visual object.

48. (original) The system of claim 44 wherein the transform comprises a skew transform for changing a perceived skew of the visual object.

49. (original) The system of claim 44 further comprising animation information associated with the transform, and wherein the animation information causes transformation data associated with the transform to change over time thereby animating the transformation of the visual object over time.

50. (original) The system of claim 40 wherein at least one function is invoked to associate a color with the visual object.

51. (original) The system of claim 40 wherein at least one function is invoked to associate gradient data with the visual object.

52. (original) The system of claim 40 wherein at least one function is invoked to associate a tile brush with the visual object.

53. (original) The system of claim 40 wherein at least one function is invoked to associate an image with the visual object.

54. (original) The system of claim 40 wherein at least one function is invoked to associate three-dimensional data with the visual object.

55. (original) The system of claim 40 wherein at least one function is invoked to associate a drawing comprising drawing primitives with the visual object.

56. (previously presented) The system of claim 40 wherein at least one function is invoked to associate audio and/or video media data with the visual object.

57. (original) The system of claim 40 wherein at least one function is invoked to associate an image effect with the visual object.

58. (original) The system of claim 40 wherein at least one function is invoked to associate a pen with the visual object, to describe how a shape is outlined.

59. (original) The system of claim 40 wherein at least one function is invoked to obtain a drawing context associated with the visual object.

60. (original) The system of claim 40 wherein at least one function is invoked to associate hit testing data with the visual object.

61. (original) The system of claim 40 wherein at least one function is invoked to associate a rectangle with the visual object.

62. (original) The system of claim 61 wherein at least one function is invoked to describe how a source rectangle should be stretched to fit a destination rectangle corresponding to the visual object.

63. (original) The system of claim 61 wherein at least one function is invoked to describe how content is positioned vertically within a container corresponding to the visual object.

64. (original) The system of claim 61 wherein at least one function is invoked to describe how content is positioned horizontally within a container corresponding to the visual object.

65. (previously presented) In a computing environment, a computer system comprising:

application program interface means in a media integration layer that is among a plurality of layers in a graphics processing environment for receiving function calls in a markup language in a native format;

high-level composition means for integrating graphics-related data and/or media-related data received via the interface means into a scene graph; and

rendering means for converting the scene graph into output that may be transmitted or displayed.

66. (original) The system of claim 65 wherein the rendering means includes low-level composition means for constructing a frame for viewing based on data received from the high-level composition engine.

67. (original) The system of claim 65 further comprising animation means, the high-level composition engine providing timeline data to the low-level composition means for interpolating the appearance of visible data across at least two frames to animate the visible data over time.